大数据下的分布式精确模糊 KNN 分类算法 *

邹劲松[1]，李　芳[2]

(1. 重庆水利电力职业技术学院 电子信息工程系，重庆 402160; 2. 重庆大学 计算机学院，重庆 400044)

摘　要：针对 K-近邻(KNN)方法处理大数据集的效率问题进行了研究，提出了一种基于 Spark 框架的分布式精确模糊 KNN 分类算法。该方法创新性地将 Spark 框架分布式 map 和 reduce 过程与模糊 KNN 结合，首先对不同分区中训练样本类别信息进行模糊化处理，得到类别隶属度，将训练集转换为添加类隶属度的模糊训练集；然后使用 KNN 算法对先前计算的类成员测试集计算得到 k 个最近邻；最后通过距离权重进行分类。针对百万级大数据集样本的实验，以及与其他算法的对比实验表明，所提算法是可行的和有效的。

关键词：大数据；分布式 Spark 框架；类隶属度；精确模糊 KNN 算法

中图分类号：TP301.6　　doi: 10.3969/j.issn.1001-3695.2018.04.0373

Accurate distributed fuzzy KNN classification algorithm for big data

Zou Jinsong[1], Li Fang [2]

(1. *Dept. of Electronics & Information Engineering*, *Chongqing College of Water Resources & Electric Engineering*, *Chongqing* 402160, *China*; 2. *School of Computer Science*, *Chongqing University*, *Chongqing* 400044, *China*)

Abstract: It has a research on the efficiency of processing large data sets with K Nearest Neighbor method, and then a distributed exact fuzzy KNN classification algorithm based on the Spark framework was proposed. The method innovatively combines the Spark framework distributed map and reduce processes with the fuzzy KNN. Firstly, the training sample category information in different partitions is processed to obtain the class membership degree. The training set is converted into a fuzzy training set with adding membership degrees, and then the KNN algorithm is used to calculate the k nearest neighbor of the previously calculated class member test set. Finally, it is classified by distance weight. Experiments on mega-scale dataset samples and comparison experiments with other algorithms show that this method is feasible and effective.

Key words: big data; distributed Spark framework; class membership degree; exact fuzzy KNN algorithm

0　引言

K-近邻方法（KNN）是一种有效的分类算法[1]，其分类规则基于对未知样本采集训练集中 k 个最相似的样本，它的相似性通常是一个距离度量，如欧几里德距离或曼哈顿距离。尽管其简单性，KNN 算法以其性能上的优势成为数据挖掘的十大算法之一[2, 3]。

KNN 算法在分类的时候给每个邻居赋予同样的重要性，因此可能存在误判的情况，越来越多的人对于 KNN 算法的改进算法作出研究。Fuzzy-KNN 通过模糊集来解决传统 KNN 的误判缺点，提高分类准确率[4, 5]，已经在医药[6]、经济和其他许多领域得到应用。KNN 算法在处理大数据集时存在运行时间和内存消耗两个主要问题[7]。而 Fuzzy-KNN 算法虽然在分类精度上高于 KNN，不过由于该算法需要一个额外的阶段来计算类的隶属度，所以时长和内存消耗会更大一些。为了解决这个问题，可以用基于 MapReduce 和 Spark 框架的云技术来解决大数据问题。

目前对于 KNN 及改进 KNN 算法对大数据处理的研究逐渐增多。文献[8]提出的方法是由两个阶段组成的近似 KNN 算法。首先，将数据分区以将整个数据集分成不同的分区；其次，计算每个分区中的 KNN，从而得到不同的原始 KNN 结果。文献[9]提出了快速 KNN 的大数据分类算法，该算法首先用聚类方法对样本进行分块，在测试过程找出距离待测样本最近的块；然后用使用 KNN 对该块（作为新的训练样本）进行分类。文献[10]提出了一种可以对大数据集进行分类的精确 KNN 方法，该方法在 map 阶段分割训练集并计算每个测试样本的 KNN；reduce 阶段收集所有候选者作为最近邻居，并报告最后 k 个最近邻居。因此，该算法能够处理运行时间较长的大训练和测试集，并获得与传统 KNN 算法一样的准确度。在文献[11]中作者通过简单的分割数据并在每个分割中应用 Fuzzy-KNN，然后收

收稿日期：2018-04-11；**修回日期：**2018-06-19　**基金项目：**重庆市教育科学“十三五”规划 2017 年度重点无经费课题（2017-GX-181）

作者简介：邹劲松（1975-），男，重庆人，讲师，硕士，主要研究方向为计算机软件与理论（zoujscq@163.com）；李芳（1979-），女，山东潍坊人，副教授，博士，主要研究方向为计算机软件与理论.

集每个分组的所有标签，并通过多数投票计算最终结果。其缺点是当训练集和测试集很大时，改算法需要太多的分割，并且在最后阶段产生大量的投票计算。

随着云计算的发展，对于大数据分类的研究成为热点，已经有文献对大数据作出研究。文献[12]文献提出基于 Storm 的流数据 KNN 分类算法，能够满足大数据背景下对流数据分类的高吞吐量、实时性和准确性的要求。文献[13]提出一种基于语言模糊规则的大数据分类系统，使用 MapReduce 框架来学习和融合规则库，能够有效实现大数据分类。文献[14]提出了一种新的大数据模糊规则分类系统，解决了随着并行度的增加，分类性能显着降低的问题。文献[15]提出了一种针对大数据的多层差分 KNN 算法，避免了传统改进算法中剪辑样本带来的判别误差，又大大降低了无效计算量。文献[16]提出了一种 MapReduce 和分布式缓存的 KNN 并行方法，提高了算法效率。

在研究了以上大数据分类方法的基础上，针对上述方法的不足，本文提出了一种在 Spark 上实现的基于精确模糊 KNN 算法(EF-KNN)。该算法利用 Spark 的内存原语通过分割数据来管理大型训练集；另外，该算法还可以通过遍历这个集合的块来处理巨大的测试集。第一阶段计算类隶属度，map 阶段分配训练集并计算每个分配的 KNN，reduce 阶段收集所有的候选者成为最近邻并获得最终 k 个最近样本，然后计算类成员并用这个信息报告一个新的训练集，加快了运行速度。第二阶段中的 map 阶段计算分配的测试样本和训练集之间的距离，每个映射得到 k 个候选并做标记。多个 reduce 阶段，通过统计 map 阶段的标记信息，计算最终 k 个邻居，然后根据先前计算的类隶属度进行分类。

1 模糊 KNN 算法

模糊 KNN 算法是对标准 KNN 算法的改进，其在精度方面非常具有较好性能。该方法与传统 KNN 的不同之处是：模糊化未知样本和 k 个近邻的距离，然后每个类别设置隶属度。该方法用训练集来预先计算类成员，之后计算测试集中每个样本的 KNN。Fuzzy-KNN 算法的正式表示法如下：

设 TR 为训练数据集，TS 为测试集，分别由确定的样本数 n 和样本 t 组成，每个样本 x_i 是一个向量 $(x_{i1},x_{i2},\ldots,x_{ij})$，其中 x_{ij} 是第 i 个样本的第 j 个特征值。TR 的每个样本都属于一个已知的类别 w，而对于 TS 来说它是未知的。模糊 KNN 有两个不同的阶段，第一个阶段是计算 TR 本身的 k_{memb} 最近邻居，通过计算 x_{train} 和 TR 的所有样本之间的距离来搜索 k_{memb} 个最接近的样本，当得到最近邻居时，就会创建如式（1）所示的类隶属度，因此训练数据集 TR 具有一个具有类隶属度向量而不是原来的类标签。

$$u_j(x)=\begin{cases}0.51+(n_j/k_{memb})\cdot 0.49 & if\ \ j=i\\ (n_j/k_{memb})\cdot 0.49 & if\ \ j\neq i\end{cases} \tag{1}$$

第二阶段计算与第一阶段类似的 k 个最近邻居，不同之处是计算 TS 的每个样本在 TR 中 k 个最近邻居，最终按式(2)进行计算。

$$u_i(x)=\frac{\sum_{j=1}^{K}u_{ij}\left(1\Big/\left|x-x_j\right|^{\frac{2}{m-1}}\right)}{\sum_{j=1}^{K}\left(1\Big/\left|x-x_j\right|^{\frac{2}{m-1}}\right)} \tag{2}$$

其中：m 为模糊权重调节因子，如果 $u_i(x)=\max\{u_z(x)\},z=1,2,\ldots,w$ 且 $w\neq i$ 时，则可以得出样本 x 属于第 i 类。

2 大数据的精确模糊 KNN(EF-KNN)算法

Spark 是 MapReduce 一个新的实现，解决了 Hadoop 的一些缺点，最重要的特征是数据的类型结构以透明的方式并行计算，它被称为弹性分布式数据集（RDDs）；另外，RDD 允许持久化和重用数据。此外，其开发与 Hadoop 合作，特别是与其分布式文件系统（hadoop distributed file system, HDFS）。

本文在 Spark 框架基础上提出了一种基于 Spark 框架大数据环境的分布式精确模糊 KNN 分类算法。该算法分为两个阶段，第一阶段计算隶属度，第二阶段进行分类。

2.1 计算类隶属度

这个过程是计算训练集类成员隶属度的 MapReduce 过程。图 1 给出了 EF-KNN 的流程。

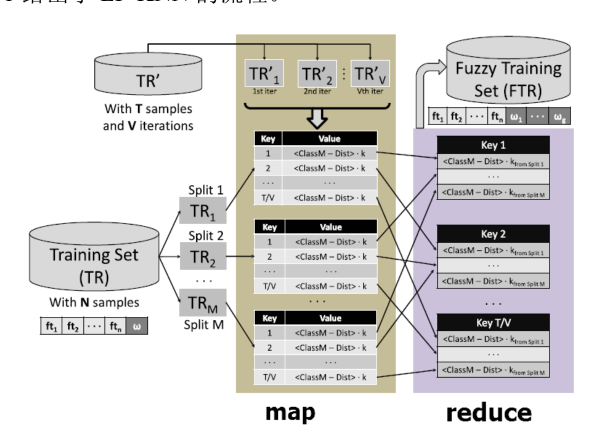


图 1 EF-KNN 流程

由图中可知，计算过程分为 map 和 reduce 部分两部分。Map 阶段划分训练数据集 TR 并计算每个分割的距离，并为每个训练样本取 k 个最近邻的类别；reduce 阶段将所有候选者加入到 k 个最近邻中，并获得 k 个最接近的定义样本，形成一个新的添加具有类隶属度向量的训练集，被称为模糊训练集（fuzzy training set , FTS）。

1）Map 过程 从 HDFS 读取的训练集 TR 作为 RDD 对象开始，TR 被分成 m 个分区，作为用户定义的参数，因此，对于每个分区 $TR_j,1\le j\le m$ 具有一个 map 任务 $(\mathrm{map}_1,\mathrm{map}_2,\ldots,\mathrm{map}_m)$，每个 map 包含大约相同数量的训练样本。

为了获得一个精确的类隶属度方法，需要对每个训练样本中的所有训练样本进行比较，以将每个训练样本与整个训练集

进行比较。假设 TR_j 和 TR 在内存中一起使用；否则，TR 将被分区成 v 块，并以顺序的方式迭代，以允许存储在内存中并正确执行。算法 1 中包含了 map 函数伪代码。

算法 1　隶属度计算中 map 函数

输入：TR_j, TR_v, k_{memb}

1、 **For** $t=0$ **to** $size(TR_v)$ **do**

2、 计算 KNN(TR_j, TR_v, k_{memb}) $\rightarrow$ $Clas\&Dist_{t,j}$

3、 (*key: t, value:* $Clas\&Dist_{t,j}$) $\rightarrow$ $result_j$

4、 **end for**

5、 输出 $result_j$

每个 map 过程建立一个 TR_v 中每一个训练样本的 k_{memb} 维<类别，距离>的向量 $Class\&Dist_{t,j}$，步骤 2 计算类别及其与 k_{memb} 最近样本的距离。为了加速还原器中最近邻居的最新实现，每个矢量 $Class\&Dist_{t,j}$ 按升序排序。每个 map 任务都会建立一个 $Class\&Dist$ 矩阵，表示候选样本是最近邻居，通过步骤 3 中过程得到。通过该方案，可以使用多个 reducers 来处理大型训练集。

2）Reduce 过程　多个 reducers 从 map 任务中输出得到的 k_{memb} 最近邻居，其目标是获得 TR_v 中包含的每个训练样本的最近邻居。 要做到这一点，所有元素都 key 分组，并计算类的隶属度，具体过程见算法 2。

算法 2　隶属度计算中 reduce 伪代码

输入：$result_{key}$, k_{memb}, $cont1=0$, $cont2=0$

While $i< k_{memb}$ **do**

　If $result_{key}(cont1).Dist \leq result_{reducer}(cont2).Dist$ **then**

　　$result_{key}(cont1).Dist \rightarrow out(i)$, $cont1++$

　else if $result_{key}(cont1).Dist= result_{reducer}(cont2).Dist$ **then if**

　　$i< k_{memb}$ **then**

　　　　$i++$

　　　　$result_{key}(cont2) \rightarrow out(i)$

　　　　$cont2++$

　　　end if

　else

　　$result_{key}(cont2) \rightarrow out(i)$

　　$cont2++$

　end if

　$i++$

end while

输出 *out*

Reduce 任务将通过合并 map 的输出来更新 k_{memb} 最近邻候选者选择，由于来自 map 的矢量根据距离排序，所以更新过程变得更快。这包括合并两个排序列表，确保尽可能地保留相同距离的邻居。这个函数一个一个地比较距离，如果距离小于当前的距离，则更新距离和类；如果距离更远，则被丢弃；如果距离完全相同，则保存两者。然后得到训练集中每个样本的 k_{memb} 近邻，另一个 map 阶段将计算类隶属度，计算方法见式(1)，并将表示每个类隶属度向量的单个标签更改。最后返回保存原始特征的每个原始样本，并更改用于计算的类成员的标签，即产生了一种新的模糊训练集，这是 2.2 节中分类部分的输入。

2.2　分类过程

图 2 给出了分类阶段的流程，分为 MapReduce 的两个基本操作。

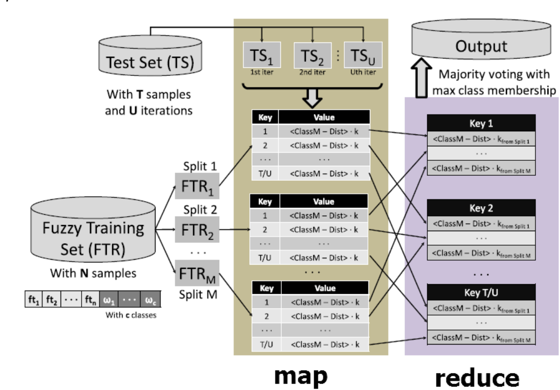


图 2　最终分类阶段流程

Map 过程需要训练数据集 *TR*，测试集 *TS* 和参数 *k* 作相应分区，并计算每个 *TR* 分区的 K 近邻和 TS 类的每一个样本的距离；reduce 阶段收集每个分区的所有 k 个最近邻，并计算 k 个最接近的样本，最后通过多数投票报告每个样本的分类标签，分类阶段的重点是获得从模糊训练集 FTR 开始的确切结果。

1）Map 过程　FTR 分成 m 个部分，其中包含大约相同数量的样本，测试集 TS 必须未分区，以便通过分布式 map 操作来计算所有候选者为 FTR 的每个分区中 k 个最近邻。如果 FTR 的分区数保持为 TR 的分区，则避免了 shuffle 阶段，因此在运行时效率会提高。

算法 3 显示了 map 函数的伪代码。对于 TS 的每个样本计算它的 *k* 个最近邻。变量 neigh-memb 保存距离和类隶属度矢量，最后将每个测试样本的 id 添加为 key，并将其发送到 reduce 阶段。

算法 3　分类过程中 map 伪代码

输入：TR_j, TS, k

1、 **for** $t=0$ **to** $size(TS)$ **do**

2、 计算 KNN(TR_j, TS, k) $\rightarrow$ $neigh\text{-}memb_{t,j}$

3、 (*key: t, value:* $neighbors_{t,j}$) $\rightarrow$ $result_t$

4、 **end for**

5、 输出 $result_t$

2）Reduce 过程　其目的是汇总所有候选最近邻，以便最终获得整个模糊训练集 FTR 的 *k* 个最近邻，保持距离和类隶属度向量。

算法 4　分类过程 reduce 伪代码

参数设置：$cand_{key,1}$, $cand_{key,2}$, k, $C1=0$, $C2=0$

While $i< k$ **do**

　If $cand_{key,1}(C1).Dist \leq cand_{key,2}(C2).Dist$ **then**

　　$cand_{key,1}(C1) \rightarrow result(i)$

　　if $cand_{key,1}(C1).Dist= cand_{key,2}(C2).Dist$

then
i++
if $i< k_{memb}$ **then**
$cand_{key,2}(C2) \rightarrow result(i)$
$C2$++
end if
end if
$C1$++
else
$cand_{key,2}(C2) \rightarrow result(i)$
$C2$++
end if
i++
end while

考虑到一些样本的距离可能是相同的，因此，当 k 参数允许时将保留两个邻居。最后，利用得到的邻居，应用最后一个 map 函数计算预测的类标签，式(2)应用于测试集的每一个样本分类。

3　实验结果与分析

采用 UCI 机器学习库的三个大数据集(PokerHand、Susy、Higgs)来评估提出的模型。表 1 给出了样本描述。

表 1　数据集描述

数据集	样本	属性数	类别
PokerHand	1 025 010	10	10
Susy	5 000 000	18	2
Higgs	11 000 000	28	2

模型中用到的参数 k_{memb} 选择 3、5、7 做邻居数量来计算类成员，k 为未知样本的邻区数，另外并发式计算 map 任务数即为数据集 TR 分区数量。

所有执行都在由另一个主节点管理的 16 个计算节点组成的集群上运行.节点具有以下特点：2 个 Intel Xeon CPU E5-2620 处理器，每个处理器 6 个内核（12 个线程），2 GHz 和 64 GB RAM，每个任务都有 2 GB 可用主内存，每个节点都使用 Spark 1.6.2 进行配置。

从运行时间和准确性方面分析了所提出的方法。表 2 给出了使用 Poker 数据集时，计算类成员阶段的时间（MembT）、分类阶段的时间（ClasT）和总运行时间（TotalT）以及分类准确度（Acc）。为了简化显示的结果，k_{memb} 和 k 的值是相同的，并且等于 3、5、7，map 数设为 256。

表 2　本文算法对 Poker 数据集实验结果

k_{memb} 和 k	MembT	ClasT	TotalT	ACC
3	462.2232	130.7489	592.9721	72.57%
5	484.2841	145.6419	629.926	73.13%
7	499.0342	141.0916	640.1258	73.36%

由表 2 中数据可以得出，对于参数 k_{memb} 和 k，由于设计的执行，尽管样本量增加以及 reduce 阶段的邻居计算量增加，但三种模型中的任何一种的总运行时间都不会增加太多，且 k_{memb} 和 k 取值 3、5、7 时分类准确率相差较少。 Susy 数据集在不同 map 数的实验结果如表 3 所示。

表 3　Susy 数据集在不同 map 数的实验结果

k_{memb} 和 k	Accuracy		
	128	256	384
3	83.38%	82.46%	82.84%
5	83.50%	82.92%	82.78%
7	83.19%	82.91%	82.38%

由表 3 中数据可知，大数据分类精度随着 map 数的增加而降低，这是因为对于测试样本，有几个训练数据点的距离完全相同，当分布式执行时，最终邻居将依赖于从每个映射输出到达的顺序。

图 3 给出了总运行时间（以 s 为单位）与 map 任务数量之间的关系，k_{memb} 和 k 的值相同且等于 3。图 4 给出了本文算法在三个数据集分类获得的运行时间（以 s 为单位），其中 k_{memb} 和 k 的值相同且等于 3 和 map 任务数量为 384。

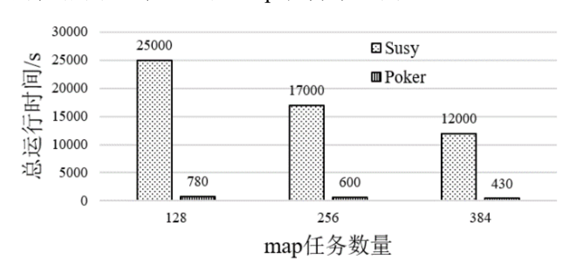


图 3　Map 数量对总运行时间的影响

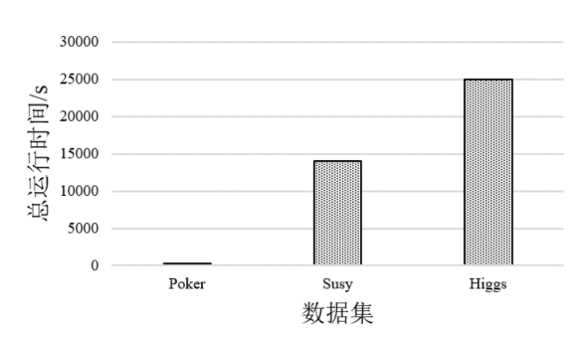


图 4　不同数据集的运行时间

分析图 3 和 4 可知，本文所提出的模型可伸缩性是线性的，但是在运行时 Higgs 数据集的时间相当高，并揭示了该算法的弱点，运行时需要更多硬件条件。

为了验证本文方法的先进性，将本文算法与其他算法进行比较。实验中本文算法中 kmemb 和 k 的值均取为 3，分类准确率对比结果见表 4。

表 4　不同算法分类准确率性能比较　　/%

数据集	方法			
	[11]	[13]	[17]	本文算法
PokerHand	75.12	71.96	77.99	89.25
Susy	69.06	60.31	63.14	83.38

Higgs	62.63	55.61	55.86	77.82

由表中数据可以得出，与其他大数据分类算法相比较，所提算法分类准确率性能优于其他算法，说明所提方法的先进性。

4 结束语

为解决大数据分类效率问题，提出一个可扩展的分布式基于 Spark 框架的精确模糊 K-NN 算法。该算法在 Spark 框架上首先进行数据分区，计算类隶属度，得到模糊数据集，最后使用 KNN 实现大规模数据的分类。实验表明，本文方法在保证分类精度的前提下，可实现百万—千万数据规模的数据分类，且总运行时间不会增加太多。通过与其他算法的实验比较，说明所提方法的可行性与有效性。未来的工作将通过一个近似模型来加速模糊 KNN 模型，着重于减少计算类隶属度时间。

参考文献：

[1] 马闯，吴涛，段梦雅. 基于 K-近邻隶属度的聚类算法研究 [J]. 计算机工程与应用，2016, 52 (10): 55-58. (Ma Chuang, Wu Tao, Duan Mengya. Clustering algorithm based on membership degree of K-nearest neighbor [J]. Computer Engineering and Applications, 2016, 52 (10): 55-58.)

[2] 郭曦，王盼，王建勇，等. 基于 K-近邻最弱前置条件的程序多路径验证方法 [J]. 计算机学报，2015, 38 (11): 2203-2214. (Guo Xi, Wang Pan, Wang Jianyong, *et al.* Program multiple execution paths verification based on K-proximity weakest precondition [J]. Chinese Journal of Computers, 2015, 38 (11): 2203-2214.)

[3] Samanthula B K, Elmehdwi Y, Jiang W. K-nearest neighbor classification over semantically secure encrypted relational data [J]. IEEE Trans on Knowledge and Data Engineering, 2015, 27 (5): 1261-1273.

[4] Kermani E F, Barani G A, Hessaroeyeh M G. Cavitation damage prediction on dam spillways using fuzzy-KNN modeling [J]. Journal of Applied Fluid Mechanics, 2018, 11 (2): 323-329.

[5] Derrac J, Chiclana F, García S, *et al.* Evolutionary fuzzy K-nearest neighbors algorithm using interval-valued fuzzy sets [J]. Information Sciences, 2016, 329: 144-163.

[6] Chen H L, Huang C C, Yu X G, *et al.* An efficient diagnosis system for detection of Parkinson's disease using fuzzy K-nearest neighbor approach [J]. Expert Systems with Applications, 2013, 40 (1): 263-271.

[7] Jivani A G, Shah K, Koul S, *et al.* The adept K-nearest neighbour algorithm: an optimization to the conventional K-nearest neighbour algorithm [J]. Trans on Machine Learning and Artificial Intelligence, 2016, 4 (1): 52-57.

[8] Deng Z, Zhu X, Cheng D, *et al.* Efficient K NN classification algorithm for big data [J]. Neurocomputing, 2016, 195 (C): 143-148.

[9] 苏毅娟，邓振云，程德波，等. 大数据下的快速 KNN 分类算法 [J]. 计算机应用研究，2016, 33 (4): 1003-1006. (Su Yijuan, Deng Zhenyun, Cheng Debo, *et al.* Fast KNN classification algorithm under big data [J]. Application Research of Computers, 2016, 33 (4): 1003-1006.)

[10] Maillo J, Ramírez S, Triguero I, *et al.* KNN-IS: an iterative spark-based design of the K-nearest neighbors classifier for big data [J]. Knowledge-Based Systems, 2017, 117: 1003-1006.

[11] Hegazy O, Safwat S, El Bakry M. A mapreduce fuzzy techniques of big data classification [C]// Proc of SAI Computing Conference. London, UK: IEEE Press, 2016: 118-128.

[12] 周志阳，冯百明，杨朋霖，等. 基于 Storm 的流数据 KNN 分类算法的研究与实现 [J]. 计算机工程与应用，2017, 53 (19): 71-75. (Zhou Zhiyang, Feng Baiming, Yang Penglin, *et al.* Research and Implementation of KNN classification algo-rithm for streaming data based on Storm [J]. Computer Engineering and Applications, 2017, 53 (19): 71-75.)

[13] Río S D, López V, Benítez J M, *et al.* A MapReduce approach to address big data classification problems based on the fusion of linguistic fuzzy rules [J]. International Journal of Computational Intelligence Systems, 2015, 8 (3): 422-437.

[14] Elkano M, Galar M, Sanz J, *et al.* A global distributed approach to the Chi *et al.* fuzzy rule-based classification system for big data classification problems [C]// Proc of IEEE International Conference on Fuzzy Systems. Naples, Italy: IEEE Press, 2017: 1-6.

[15] 耿丽娟，李星毅. 用于大数据分类的KNN算法研究 [J]. 计算机应用研究，2014, 31 (5): 1342-1344. (Geng Lijuan, Li Xingyi. Improvements of KNN algorithm for big data classification [J]. Application Research of Computers, 2014, 31 (5): 1342-1344.)

[16] 涂敬伟，皮建勇. 基于 MapReduce 和分布式缓存的 KNN 分类算法研究 [J]. 微型机与应用，2015, 34 (2): 18-21. (Tu Jingwei, Pi Jianyong. Parallelized K-nearest neighbor algorithm based on MapReduce and distributed cache [J]. Microcomputer & its Applications, 2015, 34 (2): 18-21.)

[17] 李正杰，黄刚. 基于 Hadoop 平台的 SVM_KNN 分类算法的研究 [J]. 计算机技术与发展，2016, 26 (3): 75-79. (Li Zhengjie, Huang Gang. Research on SVM KNN classification algorithm based on hadoop platform [J]. Computer Technology and Development, 2016, 26 (3): 75-79.)